

REMARKS

Claims 1, 7, and 13 have been amended. Claims 2, 8, and 14 have been deleted. No claims have been added. Claims 1, 3-7, 9-13, and 15-18 remain pending in this application. Reexamination and reconsideration of the application as amended are respectfully requested..

Attached hereto is a marked-up version of the changes made to the Specification and Claims by the current amendment. The attached pages are captioned **“Appendix A: Claims in Marked Up Form”**.

Objection to the Drawings

The Examiner objected to the drawings, in particular to Figure 1 as allegedly failing to show necessary textual labels of features or symbols as described in the Specification. Applicants have enclosed a proposed drawing correction including a table next to Figure 1 to fulfill this requirement as suggested by the Examiner. The Examiner also objected to the drawings as allegedly failing to comply with 37 CFR 1.84(p)(4) due to the reference character 12 being used to designate all workstations. Applicants have enclosed a proposed drawing correction including unique reference numerals for each workstation of Figure 1. Applicants have also amended the Specification in furtherance of these proposed drawing corrections.

Objection to the Specification

The Examiner objected to the Specification, in particular to the use of trademarks not capitalized or accompanied by the generic terminology. Applicants have amended the Specification to capitalize the use of the following trademarks, all of which were already accompanied by generic terminology: WINDOWS 95™ operating system; WINDOWS 98™ operating system; OS/2® operating system; WINDOWS NT™ operating system; MVS® data sets; IBM® LPEX editor; MVS file; MICROSOFT™ Word editor; LOTUS® WORD PRO® editor; IBM S/390® computer system. Applicants checked the entire Specification for such trademarks.

The Examiner also objected to software descriptions allegedly without software versions being specified. Applicants have amended the Specification to include software versions.

Rejections under 35 U.S.C. § 102 of Claims 1, 7, and 13

The Examiner rejected claims 1, 7, and 13 under 35 U.S.C. § 102 as being anticipated by Xu et al. (U.S. Patent no. 6,324,581). Applicants respectfully traverse this rejection for the reasons set forth below.

Applicants have amended independent claims 1, 7, and 13 to incorporate the limitations of dependent claims 2, 8, and 14 respectively. These limitations are not described or suggested by

Xu et al., and the Examiner did not assert that Xu et al. described or suggested all of these limitations.

Applicants therefore submit that the Examiner's 35 U.S.C. § 102 rejection of independent claims 1, 7, and 13 is traversed, and Applicants respectfully request that the Examiner reconsider and withdraw the 35 U.S.C. § 102 rejection of independent claims 1, 7, and 13.

Rejections under 35 U.S.C. § 103(a) of Claims 2-6, 8-12, and 14-18

The Examiner rejected claims 2-6, 8-12, and 14-18 under 35 U.S.C. § 103(a) as being unpatentable over Xu et al. (U.S. Patent no. 6,324,581) in view of Schmuck et al. (U.S. Patent no. 5,940,841). Applicants respectfully traverse this rejection for the reasons set forth below.

Relative to claims 2, 8, and 14, the Examiner asserts that Xu et al. discloses the first claim element: “determining a subset of the foreign file attributes which are equivalent to a corresponding subset of file attributes of the native file system, the subset of the foreign file attributes hereinafter known as conventional file attributes” at column 8, lines 47-59. However, Xu et al. fails to teach or suggest “determining a subset of the foreign file attributes which are equivalent to a corresponding subset of file attributes of the native file system”. At column 8, lines 47-59, Xu et al. teaches determining if a foreign file is owned by a foreign file system: “For example, when the first data mover 41 receives a file access request from its client 46, it accesses its directory of file ownership information to determine whether or not it owns the file system to

be accessed. If the first data mover 41 does not own the file system to be accessed, then the first data mover 41 sends a metadata request to the data mover that owns the file system to be accessed.”. Although Xu et al. may evaluate foreign file attributes to determine if the foreign file is owned by the foreign file system, Xu et al. does not teach or suggest determining if a foreign file attribute is equivalent to a file attribute of the native file system. Furthermore, Xu et al. does not teach or suggest determining if a subset of foreign file attributes is equivalent to a subset of file attributes of the native file system. The combination of Xu et al. and Schmuck et al. also fails to teach or suggest this first claim element, and the Examiner does not assert that combination of Xu et al. and Schmuck et al. teaches or suggests this first claim element.

The Examiner also asserts that Xu et al. discloses the second claim element: “returning the conventional file attributes to the client” at column 8, line 65 to column 9, line 6. However, Xu et al. fails to teach or suggest “returning the conventional file attributes to the client”. At column 8, line 65 to column 9, line 6, Xu et al. teaches returning foreign file attributes (metadata) from a second foreign file system (second data mover) to a first foreign file system (first data mover): “For example, if the first client 46 requests access to the second file system 44, then the first data mover 41 sends a metadata request to the second data mover 42. . . . In response to a metadata request, the data mover owning the file system . . . returns metadata including pointers to data in the file system to be accessed.”, column 8, line 54 to column 9, line 6. Thus, these teachings of Xu et al. not only fail to determine conventional file attributes, they also fail to return determined conventional file attributes to the client. The combination of Xu et al. and Schmuck et al. also

fails to teach or suggest this second claim element, and the Examiner does not assert that combination of Xu et al. and Schmuck et al. teaches or suggests this second claim element.

The Examiner also asserts that Schmuck et al. discloses the third claim element: “storing a remaining subset of the foreign file attributes which are not equivalent to a corresponding subset of file attributes of the native file system, the remaining subset of the foreign file attributes hereinafter known as extended file attributes” at column 8, lines 47 to 59. However, Schmuck et al. fails to teach or suggest this third claim element at column 8, lines 47 to 59. Schmuck et al. teaches at column 8, lines 47 to 59:

“File system implementations that support sparse files efficiently allocate disk storage only for the areas of a file to which data was written, but not for holes, or at least not for holes that are larger than the block size or the unit of disk allocation used by the file system. An index or directory based on extendible hashing is implemented using a sparse file in our preferred embodiment. Each hash bucket is stored in the file at an offset given as $i*s$, where i is the hash bucket number (starting with zero) and s is the hash bucket size (all hash buckets have the same size). The directory starts out as an empty file. When the first record is inserted, it is stored in hash bucket zero, which is subsequently written to the file, increasing the file size from zero to s .”

Thus, even if the teachings of Xu et al. and Schmuck et al. are combined, the alleged combination fails to teach or suggest the present invention as claimed by dependent claims 2, 8, or 14.

Furthermore, even if the teachings of Xu et al. and Schmuck et al. are combined with the alleged obvious modification of extracting extended file attributes from the metadata, this alleged

combination also fails to teach or suggest the present invention. Applicants therefore submit that the Examiner's rejections of claims 2-6, 8-12, and 14-18 are traversed.

Dependent claims 2, 8, and 14 have been deleted, and independent claims 1, 7, and 13 have been amended to incorporate the limitations of dependent claims 2, 8, and 14 respectively.

Applicants therefore submit that the Examiner's 35 U.S.C. § 103(a) rejection of dependent claims 2, 8, and 14 is traversed, and Applicants respectfully request that the Examiner reconsider and withdraw the 35 U.S.C. § 103(a) rejection of dependent claims 2, 8, and 14.

The Examiner also rejected dependent claims 3-6, 9-12, and 15-18 under 35 U.S.C. § 103(a) as being unpatentable over Xu et al. (U.S. Patent no. 6,324,581) in view of Schmuck et al. (U.S. Patent no. 5,940,841). Applicants respectfully traverse this rejection for the reasons set forth below. Since these claims depend from dependent claims 2, 8, and 14, and Applicants believe they have successfully traversed the Examiner's rejections of claims 2, 8, and 14, Applicants respectfully request that the Examiner reconsider and withdraw the rejections of dependent claims 3-6, 9-12, and 15-18.

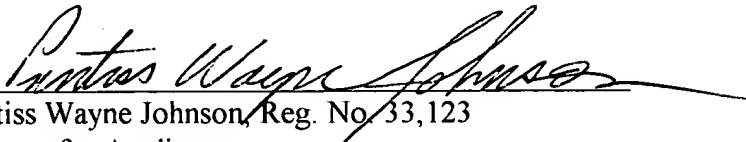
Prior Art Made of Record and Not Relied Upon

Applicant has reviewed the prior art made of record and not relied upon considered pertinent to Applicant's disclosure, and these fail to teach or suggest the claimed invention.

Conclusion

Applicants therefore respectfully request that the Examiner reconsider all currently outstanding objections and rejections and that they be withdrawn. It is believed that a full and complete response has been made to the outstanding Office Action and, as such, the present application is in condition for allowance. If the Examiner believes, for any reason, that personal communication will expedite prosecution of this Application, the Examiner is invited to telephone the undersigned at the number provided. Prompt and favorable consideration of this Response is respectfully requested.

Respectfully submitted,

By: 
Prentiss Wayne Johnson, Reg. No. 33,123
Attorney for Applicants
IBM Corporation
Intellectual Property Law
555 Bailey Avenue, J46/G467
San Jose, CA 95141-1003
(408)463-5673

Date: November 27, 2002

Appendix A: Claims in Marked up Form

1. (Amended) An extensible file access method for accessing a foreign file system from a local data processing system with a native file system, said foreign file system being located on a remote data processing system, said foreign file system having a set of foreign file attributes corresponding to each of a plurality of files in the foreign file system, and said native file system having a set of native file attributes corresponding to each of a plurality of files in the native file system, said method comprising the steps of:

- generating a request from a client on the local data processing system to the remote data processing system to open a foreign file in the foreign file system;
- opening of the of the foreign file by the foreign file system;
- sending of the file attributes of the foreign file, hereinafter foreign file attributes, to the local data processing system;
- storing of the foreign file attributes by the local data processing system;
- determining a subset of the foreign file attributes which are equivalent to a corresponding subset of file attributes of the native file system, the subset of the foreign file attributes hereinafter known as conventional file attributes;
- returning the conventional file attributes to the client;
- storing a remaining subset of the foreign file attributes which are not equivalent to a corresponding subset of file attributes of the native file system, the remaining subset of the foreign file attributes hereinafter known as extended file attributes;

20 accessing of the foreign file attributes stored in the local data processing system by the
21 local data processing system client to process the foreign file; and
22 processing by the local data processing system client the foreign file using the stored
23 foreign file attributes.

1 2. (Deleted) [The method of claim 1 further comprising the steps of:
2 determining a subset of the foreign file attributes which are equivalent to a corresponding
3 subset of file attributes of the native file system, the subset of the foreign file attributes hereinafter
4 known as conventional file attributes;
5 returning the conventional file attributes to the client; and
6 storing a remaining subset of the foreign file attributes which are not equivalent to a
7 corresponding subset of file attributes of the native file system, the remaining subset of the foreign
8 file attributes hereinafter known as extended file attributes.]

1 3. The method of claim 2 further comprising the steps of:
2 accessing of the foreign file by the client via a protocol of the native file system, the
3 accessing being performed in a similar manner to accessing a native file system file; and
4 accessing of the foreign file by the client by use of the extended file attributes, the
5 accessing being performed via a protocol different from the native file system protocol.

1 4. The method of claim 2 wherein the storing step further comprises:
2 starting an expiration timer corresponding to the extended file attributes; and
3 removing the extended file attributes from the local data processing system storage after
4 the expiration of the expiration timer.

1 5. The method of claim 2 wherein the sending of the foreign file attributes is performed by a
2 web server located on the remote system, the web server being capable of sending and receiving
3 messages via a network.

1 6. The method of claim 2 further comprising the steps of:
2 storing the extended file attributes in a shared memory portion of the local data processing
3 system storage which is accessible by the client and other local data processing system processes;
4 associating a unique handle with the extended file attributes; and
5 providing the unique handle to a local data processing system process to enable the local
6 data processing system process to access the extended file attributes.

1 7. (Amended) An article of manufacture for use in a computer system for accessing a
2 foreign file system from a local data processing system data processing system with a native file
3 system, said foreign file system being located on a remote data processing system, said foreign file
4 system having a set of foreign file attributes corresponding to each of a plurality of files in the
5 foreign file system, and said native file system having a set of native file attributes corresponding
6 to each of a plurality of files in the native file system, said article of manufacture comprising a
7 computer-readable storage medium having a computer program embodied in said medium which
8 causes the computer system to execute the method steps comprising:

9 generating a request from a client on the local data processing system to the remote data
10 processing system to open a foreign file in the foreign file system;

11 opening of the of the foreign file by the foreign file system;

12 sending of the file attributes of the foreign file, hereinafter foreign file attributes, to the
13 local data processing system;

14 storing of the foreign file attributes by the local data processing system;

15 determining a subset of the foreign file attributes which are equivalent to a corresponding
16 subset of file attributes of the native file system, the subset of the foreign file attributes hereinafter
17 known as conventional file attributes;

18 returning the conventional file attributes to the client;

19 storing a remaining subset of the foreign file attributes which are not equivalent to a
20 corresponding subset of file attributes of the native file system, the remaining subset of the foreign
21 file attributes hereinafter known as extended file attributes;

22 accessing of the foreign file attributes stored in the local data processing system by the
23 local data processing system client to process the foreign file; and
24 processing by the local data processing system client the foreign file using the stored
25 foreign file attributes.

1 8. (Deleted) [The article of manufacture of claim 7 wherein the method steps further
2 comprise the steps of:

3 determining a subset of the foreign file attributes which are equivalent to a corresponding
4 subset of file attributes of the native file system, the subset of the foreign file attributes hereinafter
5 known as conventional file attributes;

6 returning the conventional file attributes to the client; and

7 storing a remaining subset of the foreign file attributes which are not equivalent to a
8 corresponding subset of file attributes of the native file system, the remaining subset of the foreign
9 file attributes hereinafter known as extended file attributes.]

1 9. The article of manufacture of claim 8 wherein the method steps further comprise the steps
2 of:

3 accessing of the foreign file by the client via a protocol of the native file system, the
4 accessing being performed in a similar manner to accessing a native file system file; and

5 accessing of the foreign file by the client by use of the extended file attributes, the
6 accessing being performed via a protocol different from the native file system protocol.

1 10. The article of manufacture of claim 8 wherein the storing step further comprises:
2 starting an expiration timer corresponding to the extended file attributes; and
3 removing the extended file attributes from the local data processing system storage after
4 the expiration of the expiration timer.

1 11. The article of manufacture of claim 8 wherein the sending of the foreign file attributes is
2 performed by a web server located on the remote system, the web server being capable of sending
3 and receiving messages via a network.

1 12. The article of manufacture of claim 8 wherein the method steps further comprise the steps
2 of:
3 storing the extended file attributes in a shared memory portion of the local data processing
4 system storage which is accessible by the client and other local data processing system processes;
5 associating a unique handle with the extended file attributes; and
6 providing the unique handle to a local data processing system process to enable the local
7 data processing system process to access the extended file attributes.

1 13. (Amended) A distributed computer system for accessing a foreign file system from a local
2 data processing system with a native file system, said foreign file system being located on a
3 remote data processing system, said foreign file system having a set of foreign file attributes
4 corresponding to each of a plurality of files in the foreign file system, and said native file system
5 having a set of native file attributes corresponding to each of a plurality of files in the native file
6 system, said distributed computer system comprising:

7 a requestor for generating a request from a client on the local data processing system to
8 the remote data processing system to open a foreign file in the foreign file system;

9 a foreign file which can be opened by the foreign file system;

10 a sender for sending the file attributes of the foreign file, hereinafter foreign file attributes,
11 to the local data processing system;

12 storage for storing of the foreign file attributes by the local data processing system;

13 a comparator for determining a subset of the foreign file attributes which are equivalent to
14 a corresponding subset of file attributes of the native file system, the subset of the foreign file
15 attributes hereinafter known as conventional file attributes;

16 a data transfer for returning the conventional file attributes to the client; and

17 storage for storing a remaining subset of the foreign file attributes which are not
18 equivalent to a corresponding subset of file attributes of the native file system, the remaining
19 subset of the foreign file attributes hereinafter known as extended file attributes;

20 a file access for accessing the foreign file attributes stored in the local data processing
21 system by the local data processing system client to process the foreign file; and

22 a processor for processing by the local data processing system client the foreign file using
23 the stored foreign file attributes.

1 14. (Deleted) [The distributed computer system of claim 13 further comprising:
2 a comparator for determining a subset of the foreign file attributes which are equivalent to
3 a corresponding subset of file attributes of the native file system, the subset of the foreign file
4 attributes hereinafter known as conventional file attributes;
5 a data transfer for returning the conventional file attributes to the client; and
6 storage for storing a remaining subset of the foreign file attributes which are not
7 equivalent to a corresponding subset of file attributes of the native file system, the remaining
8 subset of the foreign file attributes hereinafter known as extended file attributes.]

1 15. The distributed computer system of claim 14 further comprising:
2 a file access for accessing by the client the foreign file via a protocol of the native file
3 system, the accessing being performed in a similar manner to accessing a native file system file;
4 and
5 a file access for accessing by the client the foreign file by use of the extended file
6 attributes, the accessing being performed via a protocol different from the native file system
7 protocol.

1 16. The distributed computer system of claim 14 wherein the storage further comprises:
2 an expiration timer corresponding to the extended file attributes; and
3 storage access for removing the extended file attributes from the local data processing
4 system storage after the expiration of the expiration timer.

1 17. The distributed computer system of claim 14 wherein the sender of the foreign file
2 attributes is a web server located on the remote system, the web server being capable of sending
3 and receiving messages via a network.

1 18. The distributed computer system of claim 14 further comprising:
2 a shared memory portion of the local data processing system storage which is accessible
3 by the client and other local data processing system processes for storing the extended file
4 attributes;
5 a unique handle associated with the extended file attributes; and
6 a local data processing system process which uses the unique handle to enable the local
7 data processing system process to access the extended file attributes.